STRUCTURE FROM MOTION AND DISPARITY MAP USING GMS AND LOGOS FEATURE DETECTION

Di Wang | Ahmed Nada | Neil Cai | Purrnima Dayanandam

Department of Computing and Software Systems

University of Washington, Bothell

ABSTRACT

In this project, we perform 3D reconstruction and calculate the disparity map using different matching techniques: SIFT, GMS[2], and LOGOs[6]. We experiment with these feature matching techniques and test the performance via 3d reconstruction and disparity map generation. We discovered that GMS performs better than LOGOs and SIFT in terms of correctness and efficiency.

1 INTRODUCTION

Section one, we explain the algorithms, theories and the related work of GMS and LOGOs. It also introduces the technique of Disparity Map and Structure From Motion. Section two explains how we carry out the experiment and discuss the results achieved. Section three analyzes GMS and LOGOs' performances.

1.1 GMS

J. Bian et al. [2] proposes Grid-based Motion Statistics (GMS), which "encapsulates motion smoothness" as a statistical likelihood of having a certain number of feature matches between a region pair [2]. The method focuses on separating true and false matches more reliably. The idea of this method is that a true match may have more similar matched neighbors than a false match. If motions after different camera views are smooth enough, neighborhoods around a true match should move together and should be very similar. In contrast, a false match will have less similar neighbors surrounding them. Thus, GMS counts matched features' surrounding neighbors and evaluates whether they are true matches or not.

GMS will build a grid framework for fast evaluation. Prior to the grid framework, feature points and descriptors of two images will be needed first. Then, GMS will check each match in the grid. Figure 1 shows the kernel being used to compare the neighboring cells in a feature match. In a basic scenario, the algorithm will check 8 surrounding cells and count the number of similar neighbors. If rotation is involved in the second image, the algorithm will rotate the kernel and check the surrounding neighbors. The kernel will be rotated 8 times, which covers 360 degrees. If the second image has a different scale, the algorithm will use 5 different cell sizes for the grid framework in the second image. In the code implementation, rotation and scale check need to be activated by two parameter values in the function call. GMS itself can't detect rotation and scale difference between two images.

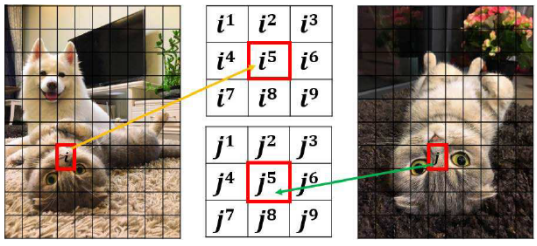


Figure 1: *GMS Predefined Grid and Kernels* [2]

1.2 LOGOs

LOGOs is short for Local Geometric support for high outlier spatial verification. The paper [6] proposes a method to determine inliers for the purpose of visual localization. For the purpose of visual localization, a visual bag of words approach like VLAD descriptors are built for images in a database using features

detected by descriptors like SURF. Query image descriptors that are assigned to a visual word are considered as potential inliers to the source image's descriptors assigned to the same visual word. These potential inliers are filtered using LOGOs or RANSAC.

The advantage of this method is the robustness when the proportion of outliers is higher. The use of local geometric properties in the neighborhood of feature points is used for determining inliers.

The proposed method shows significant improvement in Recall at 100% precision metric over multiple datasets for the visual localization task, and the ablation study on the effect of outliers on LOGOs vs RANSAC also shows how effective LOGOs is when there are more outliers. The paper also reports computational requirements for their method to support the fact that it is practically useful. The method however does not provide significant improvements for pose estimation, and only performs marginally better as compared to RANSAC.

1.3 Disparity Map

A disparity map is nothing but a depth map, in which an image contains depth information of every pixel stored in it. The term disparity in stereo vision refers to the apparent shift in pixel or motion in a pair of stereo images. To explain this concept better, let us do a quick experiment. Step 1: hold an object in front of your eyes. Step 2: close your eyes. Step 3: open the left eye, keep the right eye closed and observe the object, and then at the same time quickly open your left eye and close your right eye. When you do this, you notice a shift in the object, this is disparity. Quick fact check, objects far from the eyes shift less, and objects nearer shift more! This means that there is an inverse relationship between disparity and depth. In fact depth and disparity are related by the following equation:

$$D = X_L - X_R = (f\,(X_R+b) - X_R))/Z = fb/Z$$

where X_L and X_R are the x-coordinate of a point in the left image and right image respectively of a rectified stereo image pair. ff is the focal length of the camera, b is the baseline, and Z is the depth of that point in 3D space.

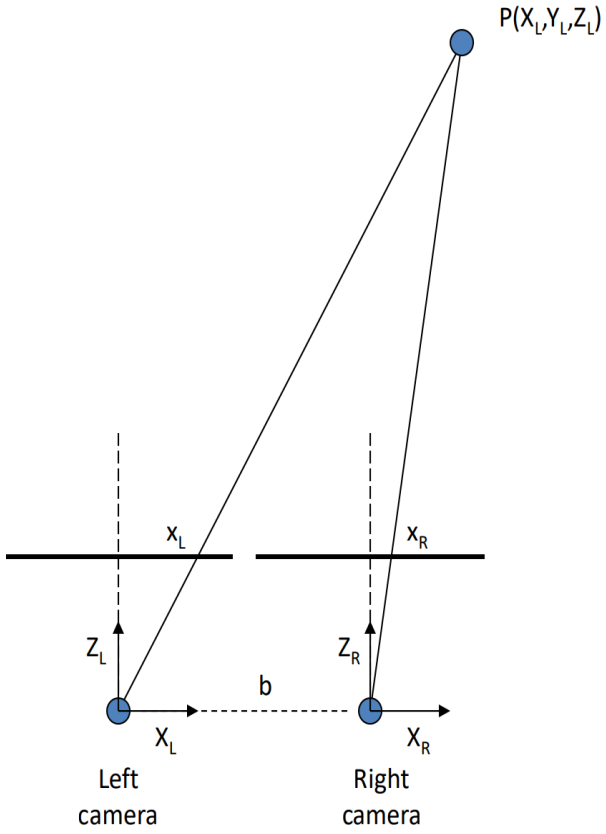


Figure 2: Stereo Geometry

1.4 Structure from Motion

Structure from Motion (SfM) is a technique which utilizes a series of 2d images to reconstruct the 3d structure of a scene. SfM produces point cloud based 3-D models similar to LiDAR. [7]

To create a 3-d reconstruction one simply needs many images of an area or an object with a high degree of overlap, taken from different angles. The camera does not need to be specialized; standard consumer-grade cameras work well for SfM methods. The images can often be taken from a moving sensor (UAVs for example) but can also be taken by a person or multiple people at different locations and angles.

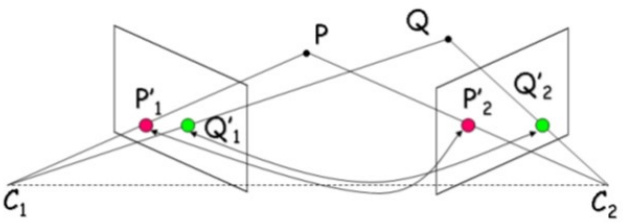
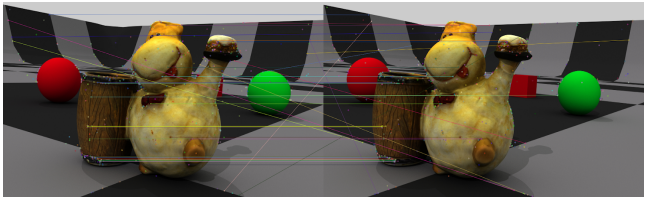


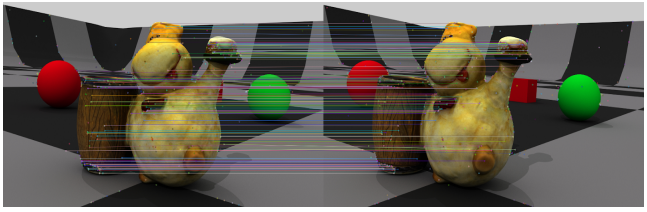
Figure 3: The triangulation process determines the 3d position of the point. [7]

2 Experimentation and Result

2.1 Feature Match

We tested the three feature matchers in three different scenarios, which are basic scenarios with a slightly different camera view, 180-degree rotation on the basic scenario, and scale changes on the basic scenario. In Figure 4.1, 4.2, and 4.3, the top figure is the result from SIFT BF matcher and the middle figure is the result from GMS, and the bottom figure is the result from LOGOs. GMS and LOGOs did a good job on eliminating the false match in the BF matcher and detecting more matches in Figure 4.1 and Figure 4.2. LOGOs do not seem to work really well when the scale becomes different, which is indicated in Figure 4.3. LOGOs have way fewer true matches compared to GMS in Figure 4.3. In Figure 4.1 and Figure 4.2, LOGOs still have a small number of false matches. In general, GMS has returned the best results so far and the run time is significantly shorter when executing GMS.





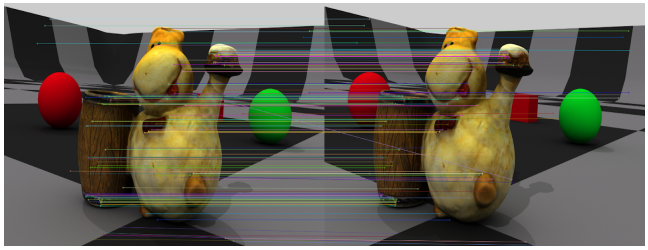


Figure 4.1: GMS vs. SIFT Basic Scenario

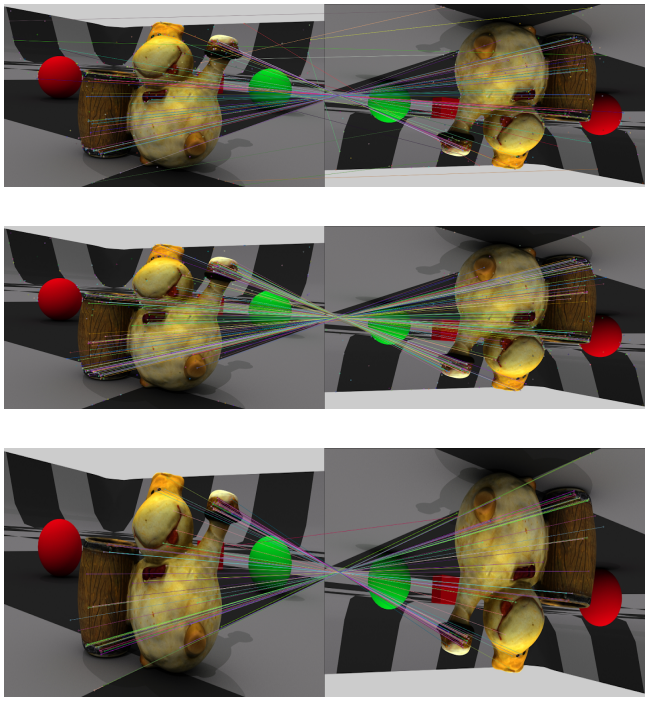
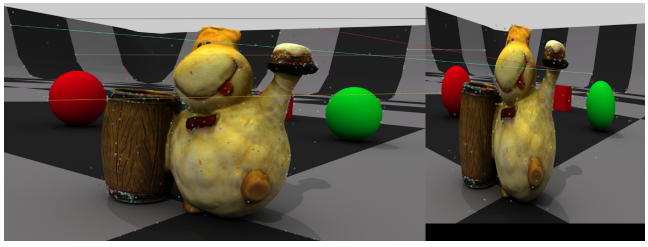
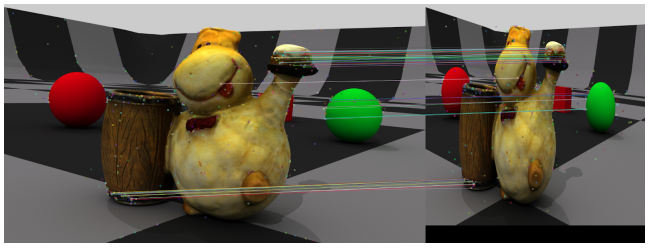


Figure 4.2 GMS vs. SIFT Rotation Scenario





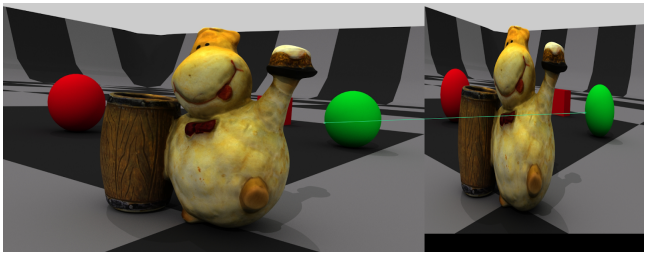


Figure 4.3 GMS vs. SIFT Different Scale Scenario

2.2 Structure from Motion

The fundamental feature extraction of SfM is based on SIFT, but LOGOS and GMS provide different match algorithms to eliminate outliers.

We calibrated our cell phone camera to calculate the intrinsic Matrix. Using this intrinsic Matrix and corresponding points, we could calculate the 3d coordinates of the matching points via triangulation.

To compare the effectiveness of LOGOS and GMS, we measured the density and correctness of the 3d cloud points against the default Brute Force Match.

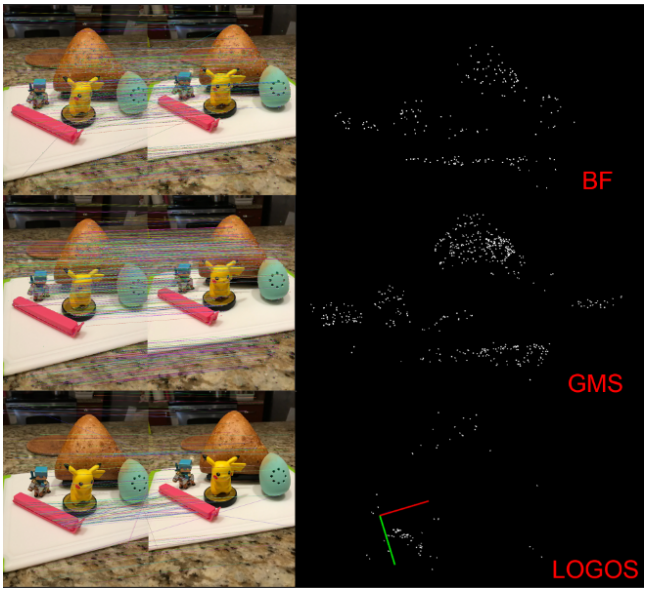


Figure 5: Feature Match and point cloud generation using SIFT+BF (top), SIFT+GMS (center) and SFIT+LOGOs (lower).

2.3 Disparity Map

In this paper we use feature matching to calculate disparity maps and compare between different feature matching algorithms as well as the feature outliers elimination algorithms applied to them.

There are two types of disparity maps, dense and sparse disparity maps. In dense disparity maps, we used as many features as possible by selecting all image pixels as keypoint candidates. Sparse disparity maps will be generated using the limited number of feature descriptors and matches that each algorithm can detect. The disparity maps depend on feature matching, we used SIFT, ORB, SIFT + GMS and SIFT + LOGOs.

To compare the results, we compare the resultant disparity maps to a ground truth. Therefore, we can calculate the Root Mean Square difference between the pixel values of the disparity maps and the ground truth.

2.3.1 Dense Disparity Map

For a dense disparity map, we consider each corresponding pixel between two input images. The result should expect the inlier feature matches to be horizontal lines.

Below is an example of a pair of stereo images:

Figure 6: Stereo pair of images(Left and Right)

We have decided to calculate the disparity map for this pair of images using feature matching. The resulting dense disparity map is shown below.

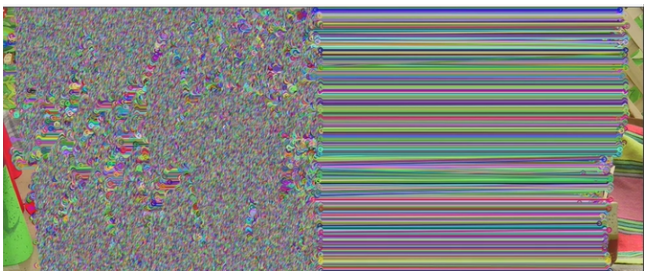


Figure 7: The result should expect the inlier feature matches to be horizontal lines

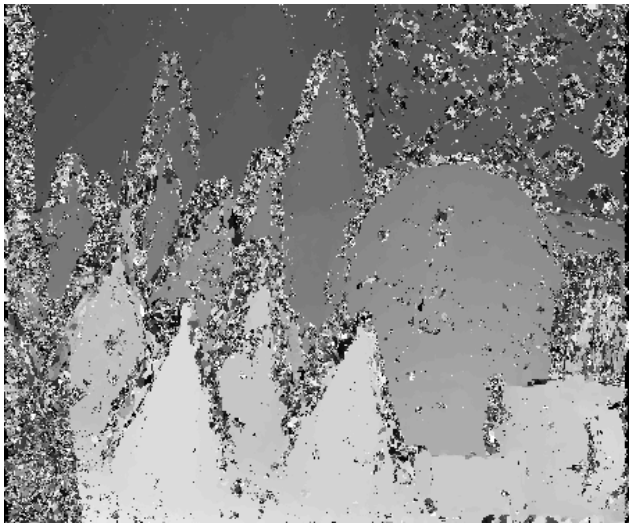


Figure 8: Feature Matching generated "Dense" disparity map

From the above resulting disparity map, it is noticed that objects that are closer to the camera taking that shot, have relatively higher disparity and so a brighter color (closer to white). However, the distant objects in the background have not moved much so the disparity is close to zero; therefore features of objects detected in the front will have brighter values closer to white or gray vs objects in the background with disparity closer to zero and so will have black color.

2.3.2 Sparse Disparity Map

Sparse disparity map uses only on the detected features. Most likely, there are a limited number of keypoints detected. Hence, the resultant disparity map contains less pixel disparities.

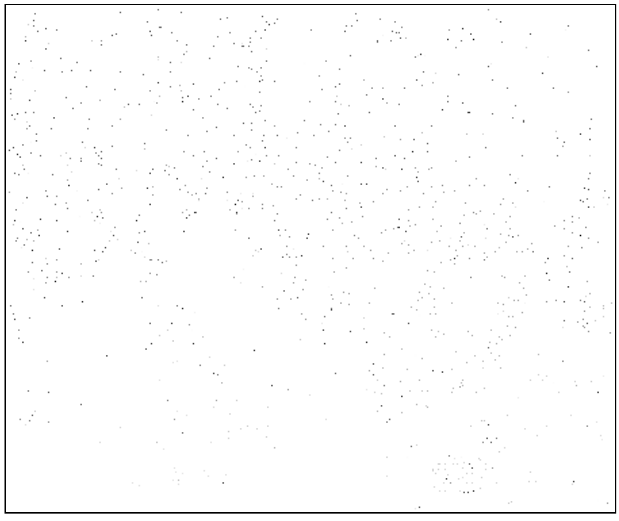


Figure 9: Feature Matching generated "Sparse" disparity map

2.3.3 Portrait Mode

We use disparity maps to generate Portrait mode. As the disparity map contains the depth information in black and white, we can translate the disparity values into blurriness.

Below are the pair of input images we have used to create a portrait mode.

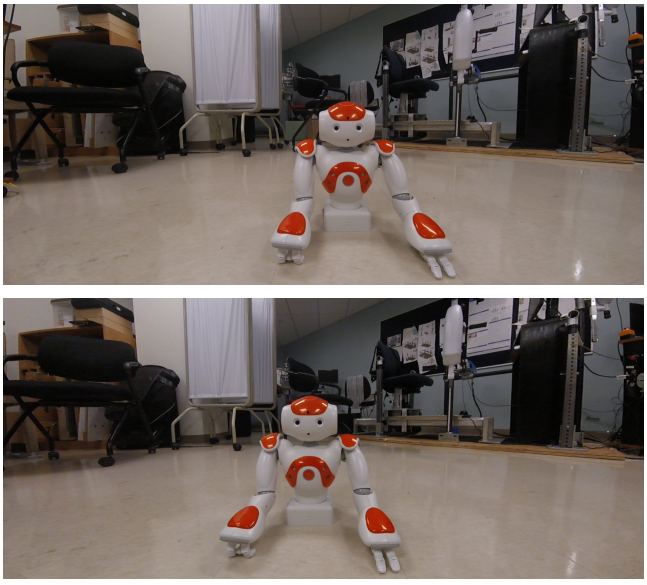


Figure 10: Pair of stereo images

The following flow-chart describes the process we followed to generate the portrait mode from a pair of stereo images.

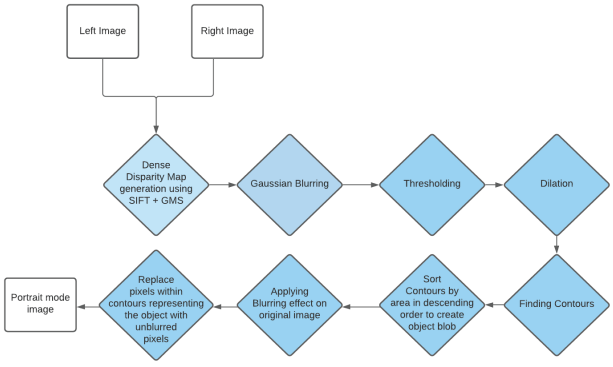


Figure 11: Portrait Mode generation flow chart

Next we generate the dense disparity map for that pair of stereo images. The dense disparity map indicates the distance of the object to the camera. Then we turned denoise by applying it to GaussianBlur and then applied a threshold function. Then we apply threatholding add more contrast to the output. Below is how the disparity map looks like after applying the Gaussian Blurring and thresholding.

Figure 12: Disparity map after Gaussian Blurring and thresholding

Next we applied a dilation function to the output. We use a discovery function to detect contours. By sorting the contours we can select the best 8 largest ones to define the body of the robot.

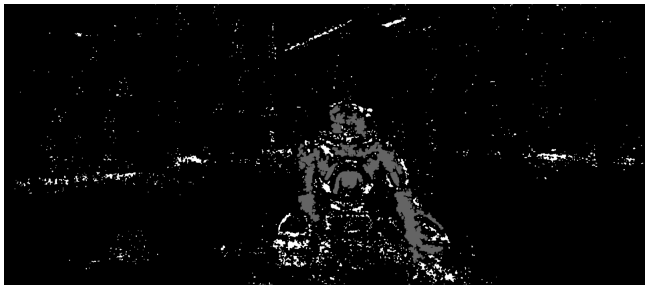


Figure 13: Disparity map after selecting the contours

After selecting the contours, the pixels representing those contours were kept in the original image while the remaining pixels were blurred using Gaussian Blur to show the resulting image as below.

Figure 14: Resultant portrait mode based on dense disparity map

3 Performance Evaluation

3.1 Structure from Motion

The result shows that GMS provides faster matching time and generates more cloud points. GMS requires

only half of the default Brute-Force (BF) Matching time but generates 617 cloud points. That is a triple of the BF's output. What surprises us is that LOGOS requires 30 seconds to match the SIFT feature points, but only generates 88 cloud points. One thing to notice that the 88 cloud points are not an accurate representation of the 3d structure.

	SIFT+BF	SIFT+ GMS	SIFT+LOGOs
Matching time (sec)	8.849	3.658	30.16
Number of cloud points	214	617	88

Table 1: Comparison of GMS and Logos.

3.2 Sparse Disparity Map

Table-2 shows that GMS has the second least RMS error compared to the ground truth image, second to LOGOs. However, GMS has around 21% more disparities than LOGOs. GMS also runs faster than all of the other two algorithms we have tried.

Algorithm	Time taken (seconds)	RMS compared to ground truth image	Number of Disparities
SIFT	0.22	71.82	1033
ORB	0.35	55.12	420
SIFT + LOGOs	0.45	6.1	326
SIFT + GMS	0.21	6.48	396

Table 2: Comparison of SIFT,ORB, SIFT+LOGOs and SIFT+GMS

3.3 Dense Disparity Map

As shown in Table-3 below, GMS has the best performance in generating the dense disparity map. SIFT + GMS was able to get 114k good matches which resulted in only 5.74 RMS error compared to the ground truth disparity image. GMS also did take the shortest time to generate the output compared to SIFT and ORB.

We have decided to eliminate LOGOS from the dense disparity map since it takes a lot of time computing the clustering for each pixel and ends up with an inaccurate dense disparity map.

Algorithm	Time taken (seconds)	RMS compared to ground truth image	Number of Disparities
SIFT	10.14	50.23	168683
ORB	35.16	35.51	121422
SIFT + GMS	9.33	5.74	113804

Table 3: Comparison of SIFT, ORB and SIFT+GMS

CONCLUSION

In conclusion, GMS generally performed better than LOGOs and SIFT in terms of accuracy, efficiency, and robustness in the implementation of 3d reconstruction and disparity map. GMS returns the feature matching results very quickly while having more accurate matching. High accuracy and robustness provided by GMS allows disparity map and 3D reconstruction to return better results. Disparity map has less noise and 3D reconstruction can generate more points. In contrast, LOGOs and SIFT have more noise in the disparity and less points in the results of 3D reconstruction. In some cases, LOGOs may even perform worse than SIFT.

REFERENCES

[1] O. Enqvist, F. Kahl and C. Olsson, "Non-sequential structure from motion," *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, 2011, pp. 264-271, doi: 10.1109/ICCVW.2011.6130252.

[2] J. Bian, W. Lin, Y. Matsushita, S. Yeung, T. Nguyen and M. Cheng, "GMS: Grid-Based Motion Statistics for Fast, Ultra-Robust Feature Correspondence," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2828-2837, doi: 10.1109/CVPR.2017.302.

[3] M. Geppert, Privacy Preserving Structure-from-Motion. ECCV 2020.

[4] “Experimental 2D Features Matching Algorithm,” *OpenCV*. [Online]. Available: https://docs.opencv.org/master/db/dd9/group__xfeatures2d__match.html. [Accessed: 01-May-2021].

[5] O. Padierna, “Stereo 3D reconstruction with openCV using an iPhone camera. Part I.,” *Medium*, 10-Jan-2019. [Online]. Available: https://becominghuman.ai/stereo-3d-reconstruction-with-opencv-using-an-iphone-camera-part-i-c013907d1ab5. [Accessed: 01-May-2021].

[6] C. Lu, L. Xu, J. Jia “LOGOs: Local geometric support for high-outlier spatial verification” IEEE International Conference on Robotics and Automation, 2018 https://github.com/short-circuitt/LOGOs/blob/master/LOGOs.pdf

[7]Structure From Motion

https://gsp.humboldt.edu/OLM/Courses/GSP_216_Online/lesson8-2/SfM.html

[8] C.-H. Lee and D. Kim, “Dense disparity map-based pedestrian detection for intelligent vehicle,” in *2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore, Singapore, Aug. 2016, pp. 108–111. doi: 10.1109/ICITE.2016.7581317.